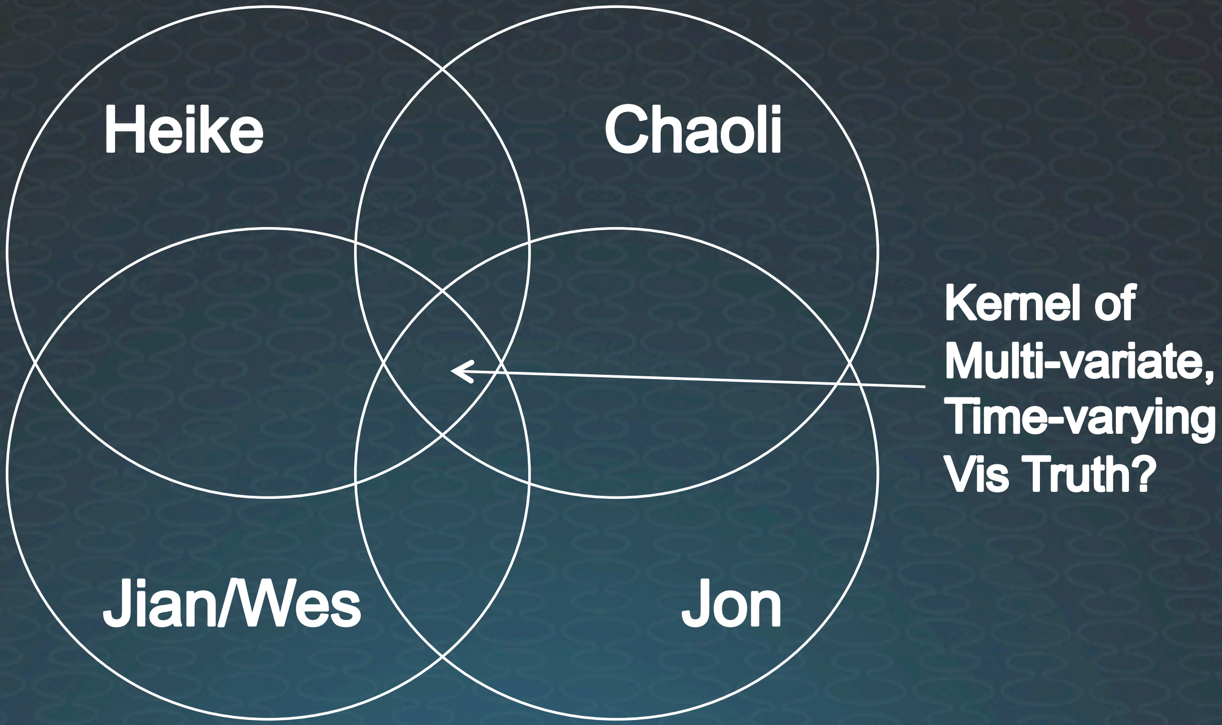
Comparative Visualization and Trend Analysis Techniques for Time-Varying Data

Jon Woodring
Los Alamos National Laboratory

VisWeek 2009

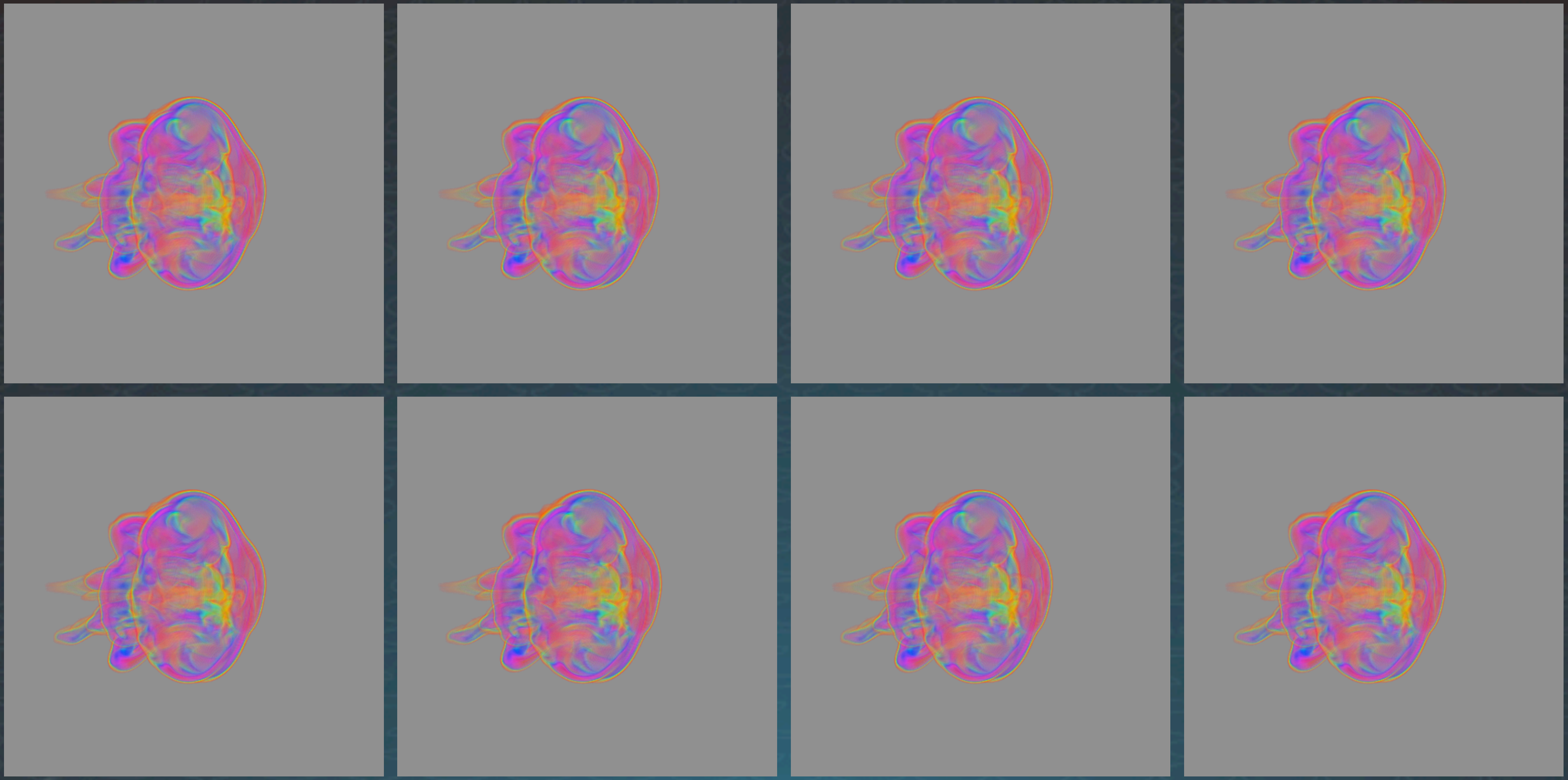
I was just noticing…



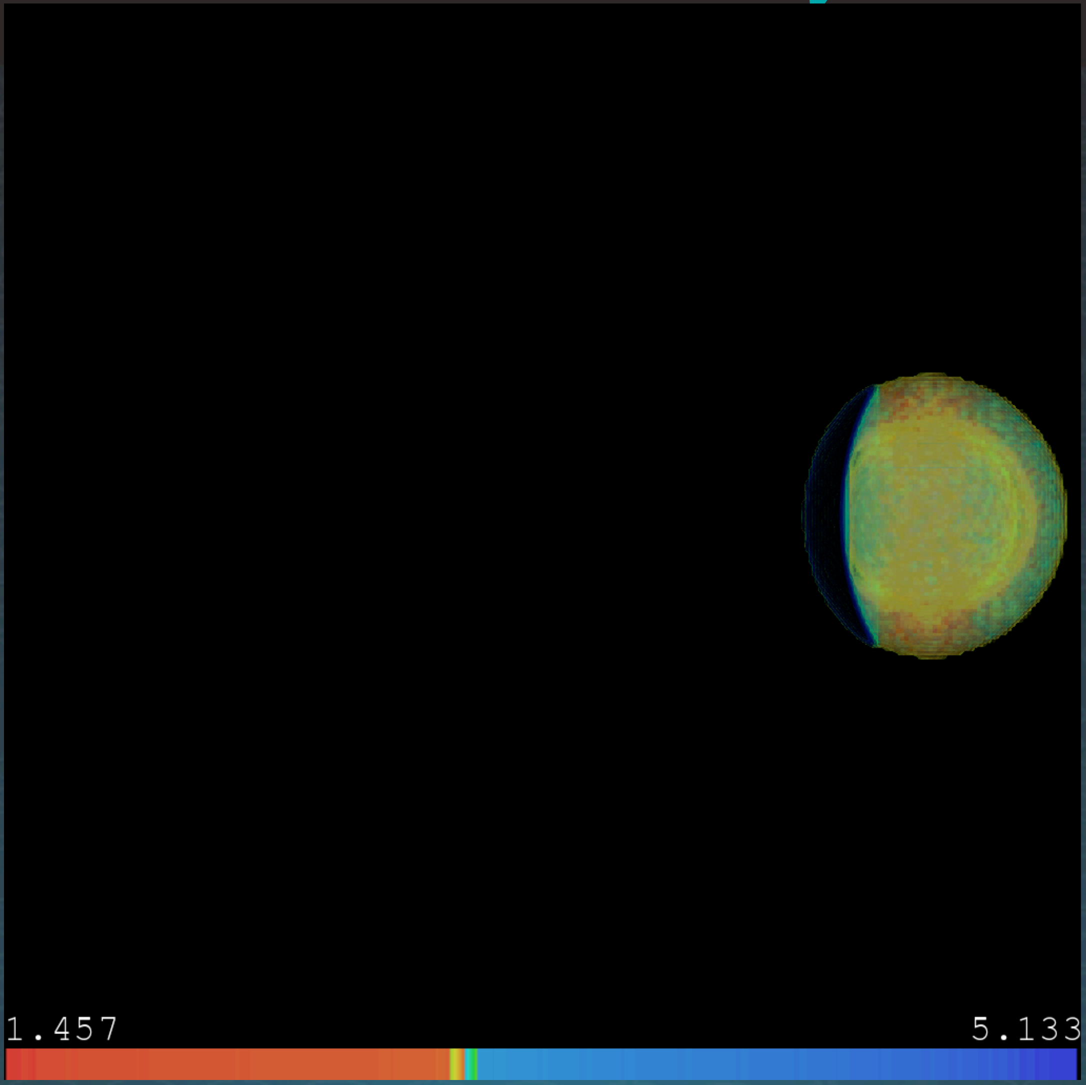
Problem Statement

- Time varying visualization for scientific data has typically been done with animation and/or time step still renders
- Animation or frame comparison may not be the only way to understand time-varying data
  - Perceptual, visual, and cognitive issues
  - Lack of knowledge of temporal trends
  - Hard to make a transfer function for time data

Locating Differences – The Worst Case



Animation –
Short Term Visual Memory



Count the Passes

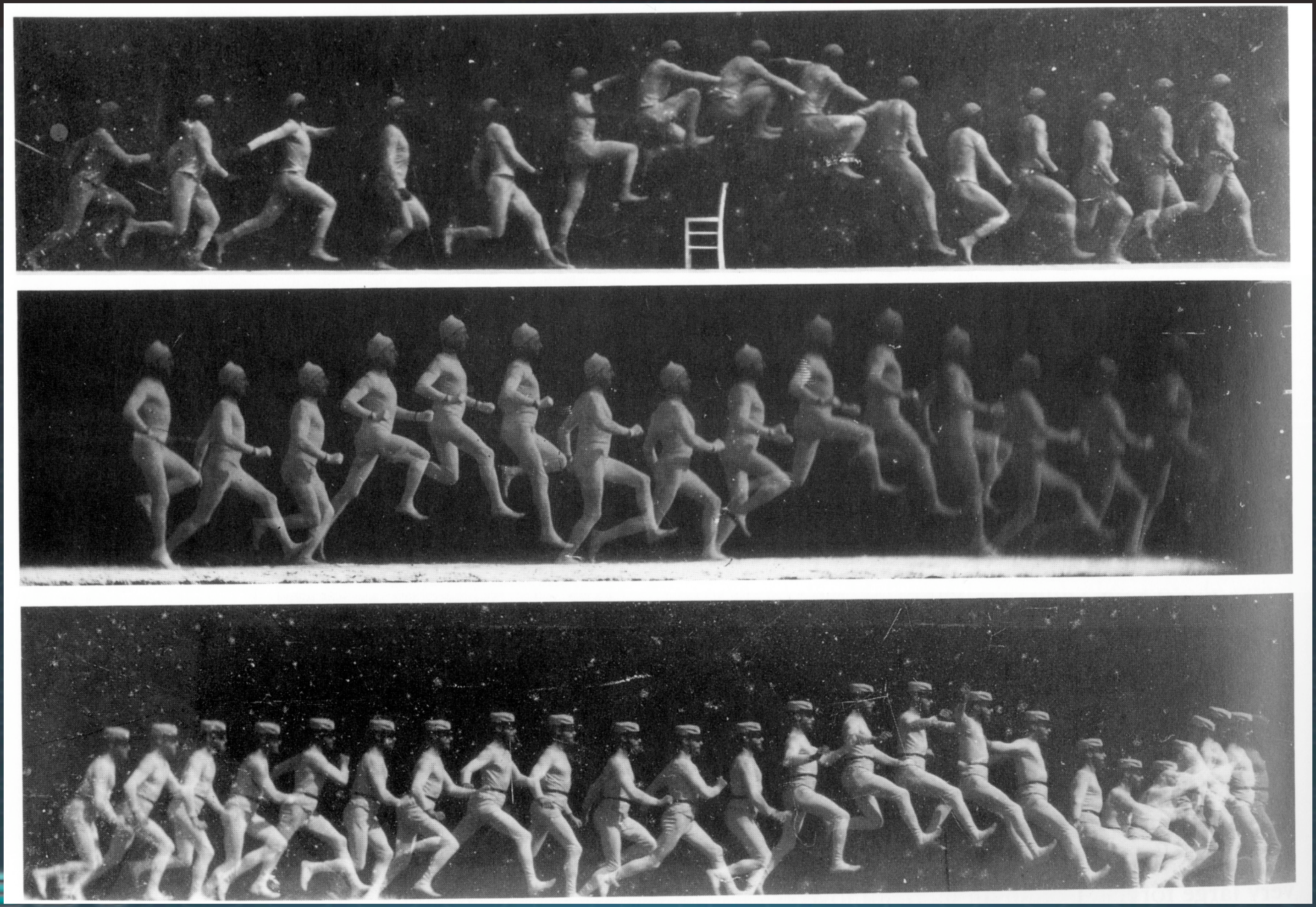
Lack of Quantitative Knowledge – Classifying Time Data

- What values does the time series have over time?
  - What are the value ranges over a time period?
  - What data points or features share similar value trends or are different?
- Transfer functions (classifying data/features) are hard
  - Tons of literature for just for making transfer functions for single time steps
  - Lack of knowledge of changing values and trends…
  - What values and data points should we classify?
  - How to classify them over time?

Approaches

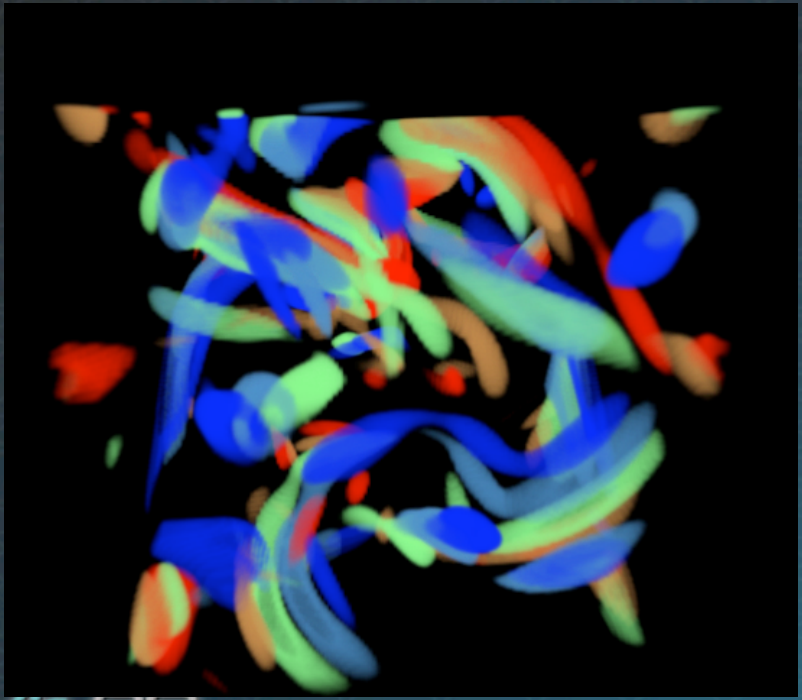
- Comparative Visualization
  - Single frame “fused” comparisons of multiple time steps
  - Visually compare changes over time in space and value to find temporal features
- Trend Analysis
  - Visualize temporal value trends in a data set for quantitative assessment
  - Computationally analyze temporal trends to extract features for classification

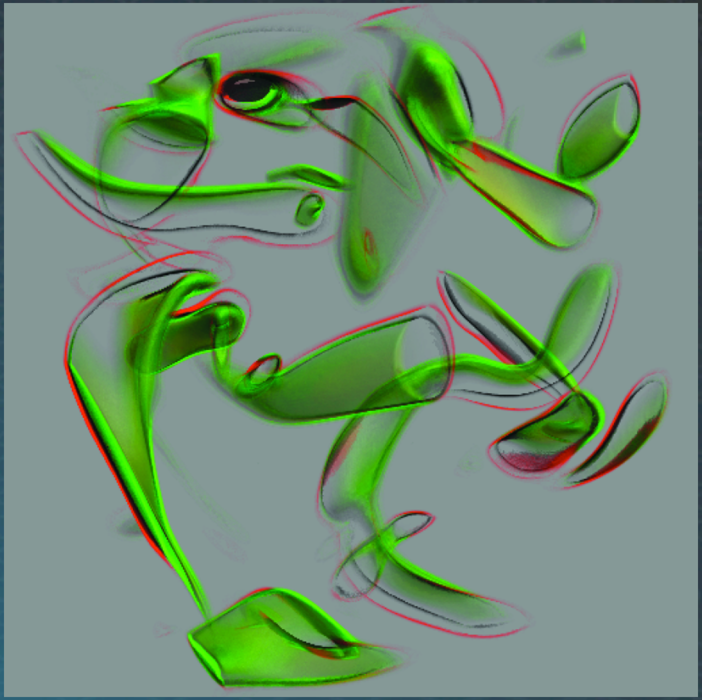
Comparative Visualization



Comparative Visualization

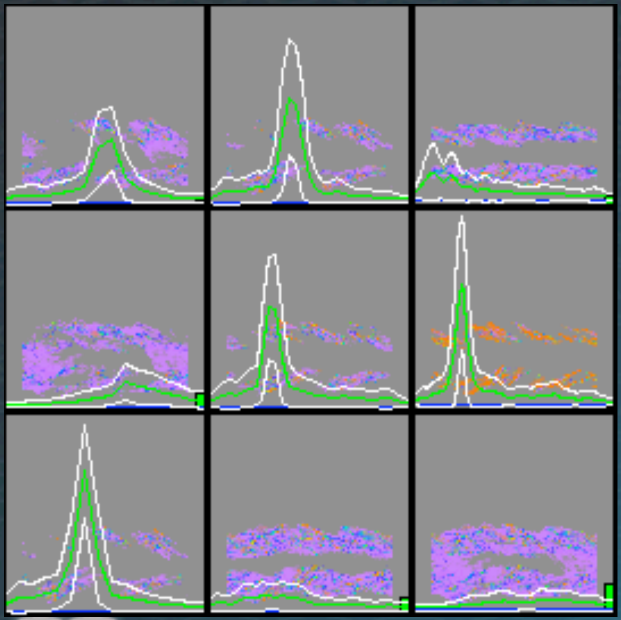
- Combine multiple time steps into a single static data set

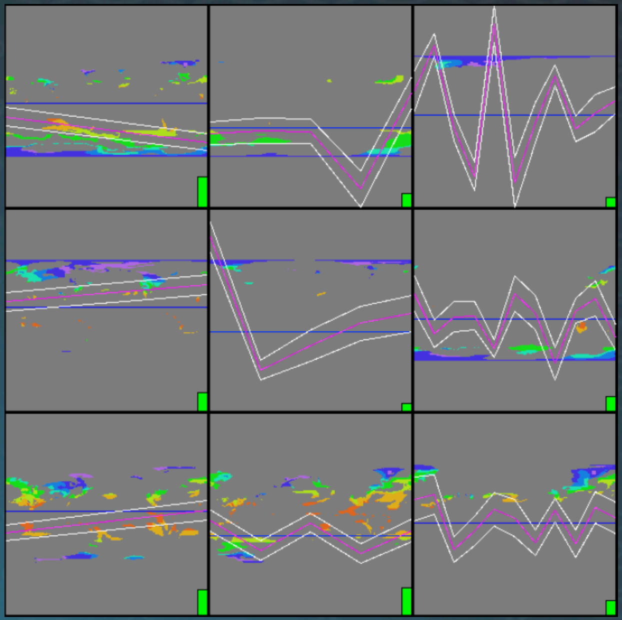




Trend Analysis

- Value activity representation allows for quantitative trend knowledge

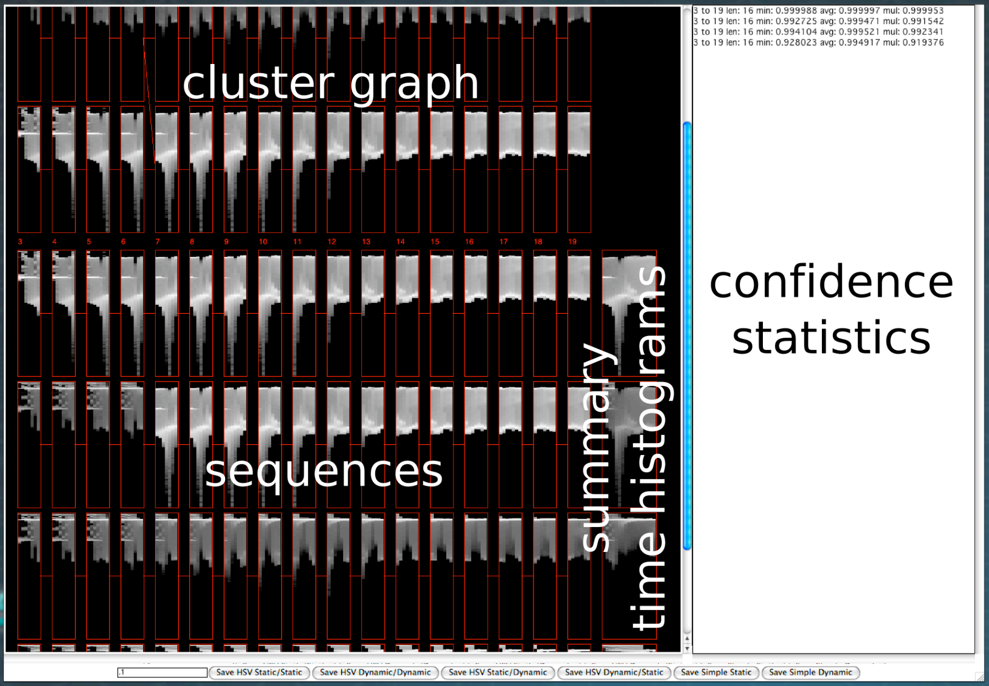




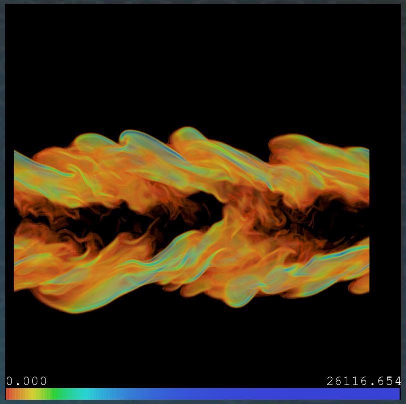
Vis09

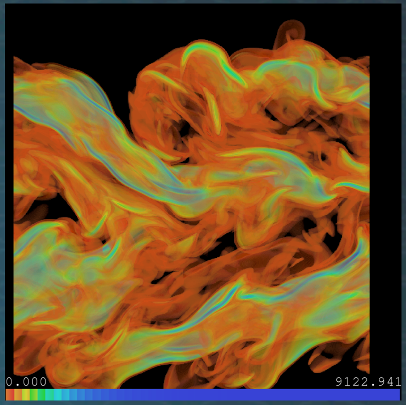
Trend Analysis

- Computationally analyze value trends for classification and feature definition



```
3 to 19 len: 16 min: 0.999988 avg: 0.999997 mul: 0.999953
3 to 19 len: 16 min: 0.992725 avg: 0.999471 mul: 0.991542
3 to 19 len: 16 min: 0.994104 avg: 0.999521 mul: 0.992341
3 to 19 len: 16 min: 0.928023 avg: 0.994917 mul: 0.919376
```

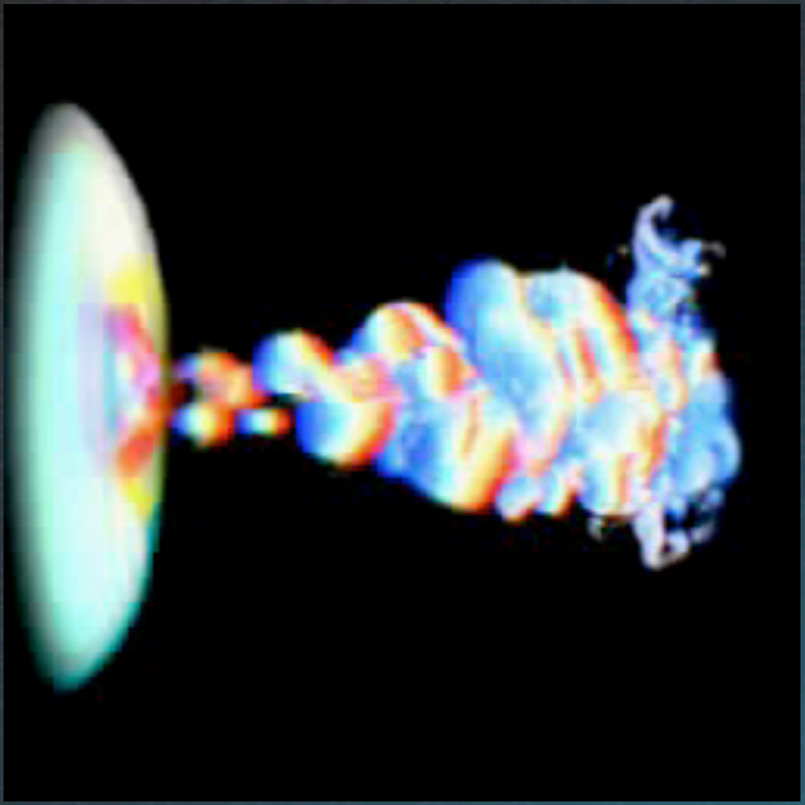


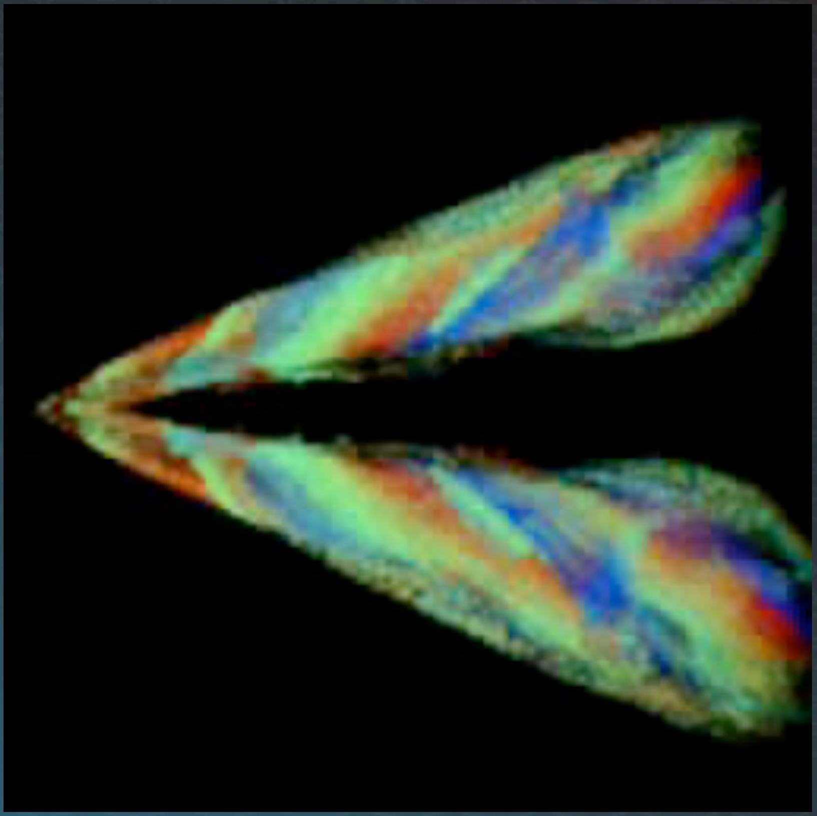


Outline

- Comparative Visualization
  - Chronovolumes
- Trend Analysis
  - Multi-scale temporal trend spreadsheet
  - Time histograms
  - Semi-automatic temporal transfer functions

Comparative Visualization - Chronovolumes

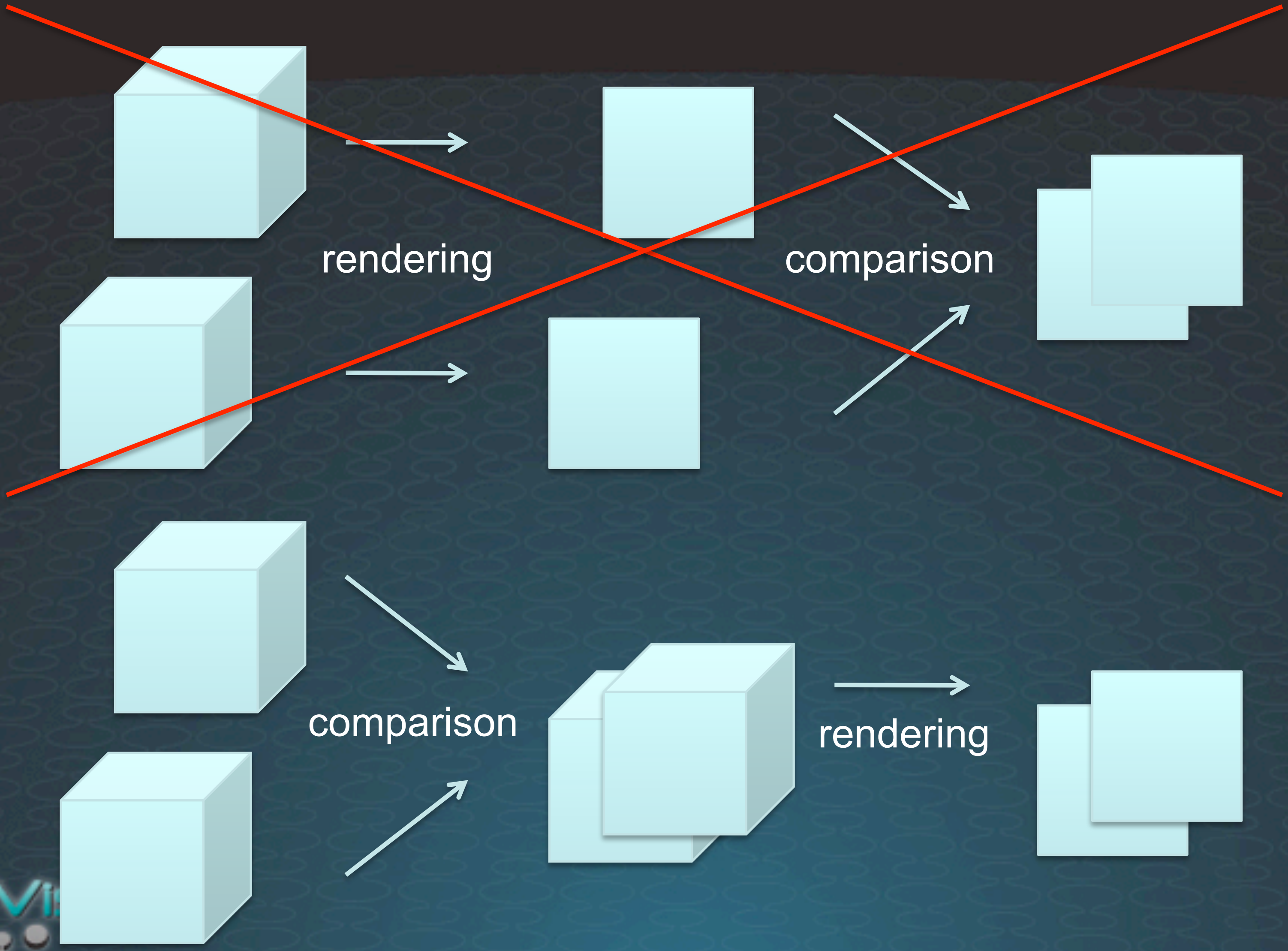




Chronovolumes

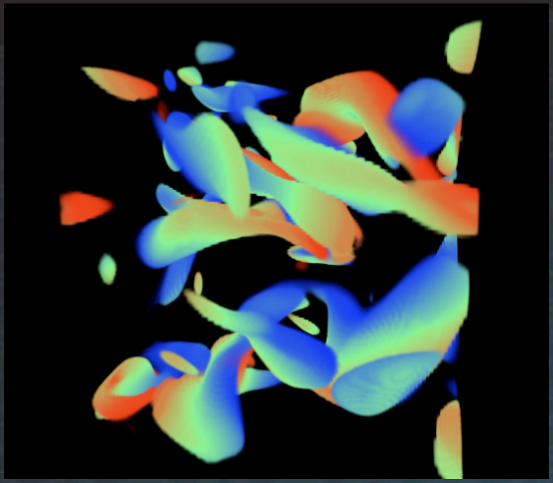
- Visually compare time series data by visually fusing several time steps into one volume
  - Provides temporal context into the frame
  - Full 3D comparison, not image composition
    - Image pixel comparison is not the same as data point comparison
    - Spatial position preservation in the comparison
    - Compare per data point over time basis, not per projected point

Chronovolumes

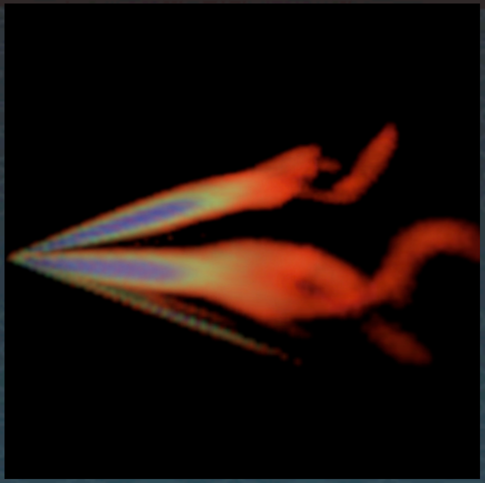


More Chronovolume Examples

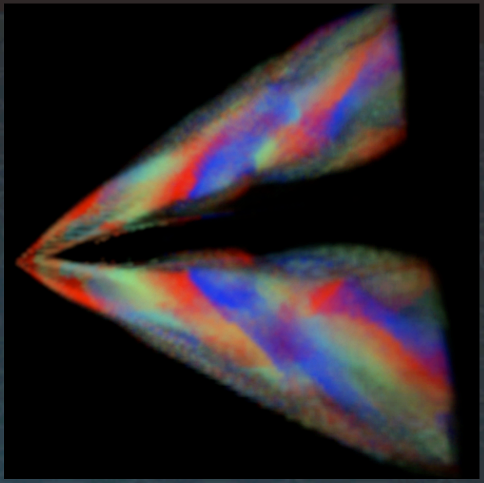
Alpha composition



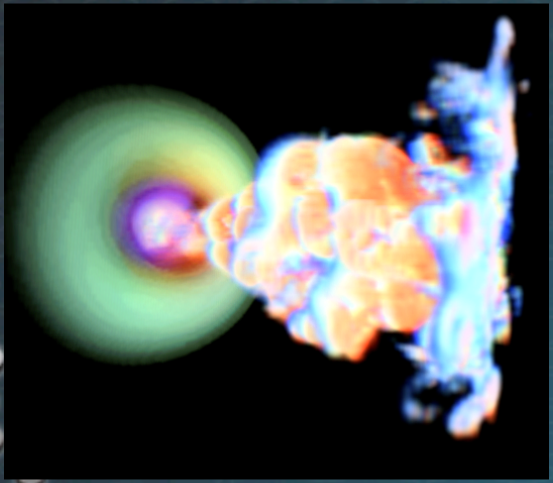
Average



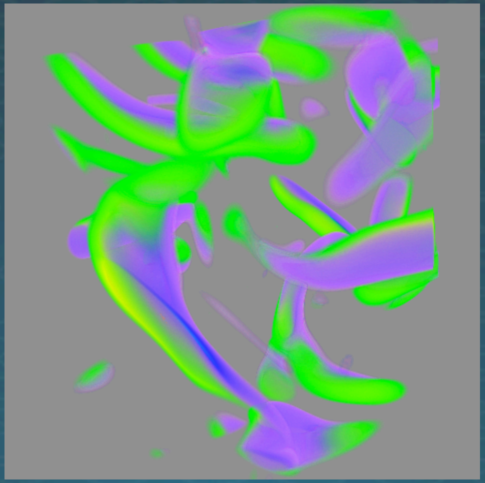
Min



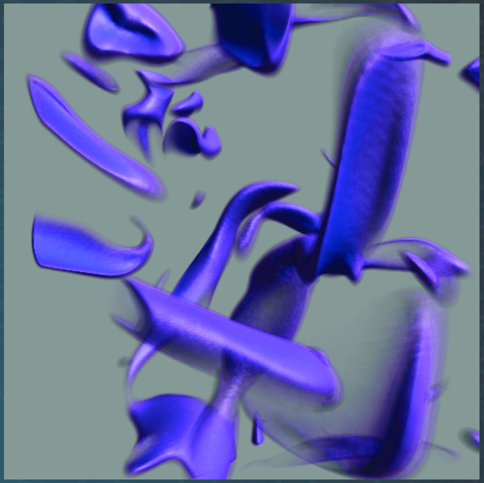
Additive color



XOR



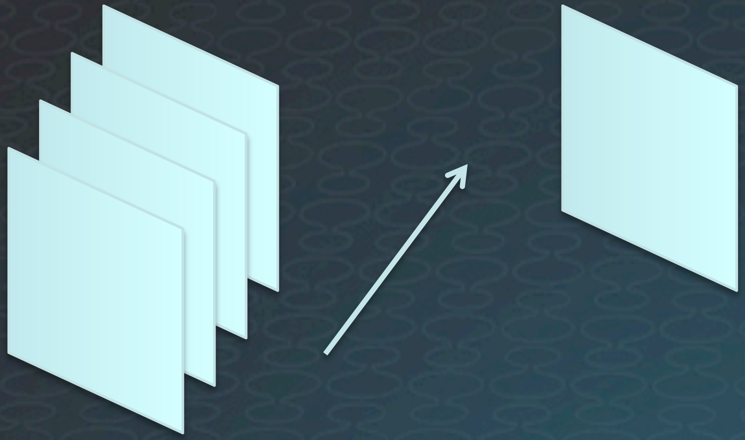
OUT



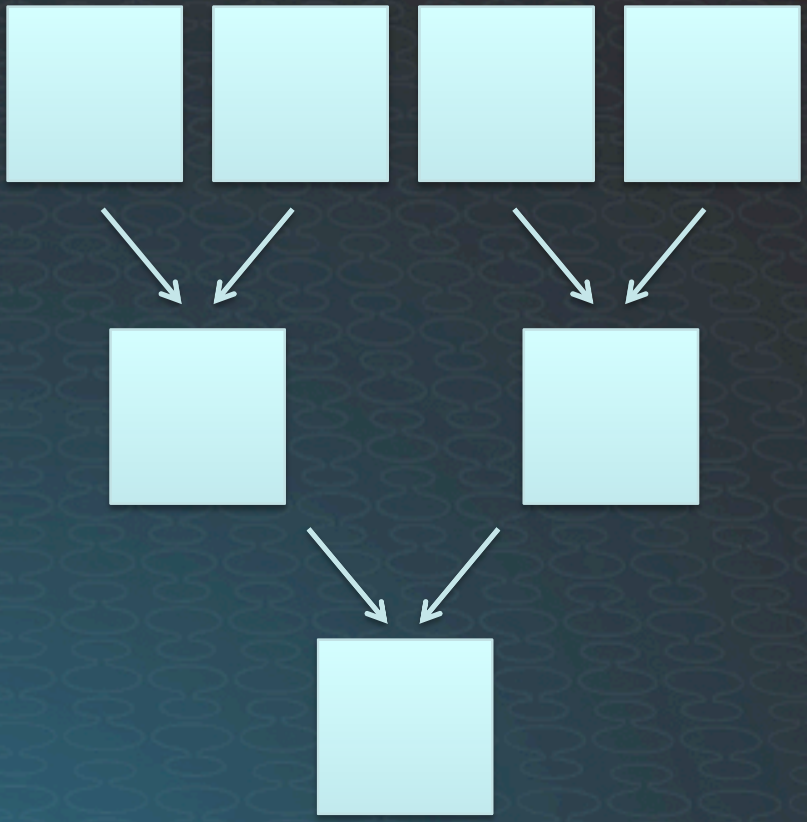
Comparison Implementations

- High Dimensional Projection (reduction)
  - Treat the time-varying data as 4D (space + time) data and project down to 3D, apply operations per data point over time
- Composition (arbitrary comparison)
  - Compose several 3D volumes together into one volume, with operation trees (kernels) per data point
- Both of these are massively data parallel operations – can be easily implemented in a volume renderer/shader (GLSL, OpenCL, Cuda, MPI, C/C++ threads, etc.)

2D Analogy – Arrow is operations applied in parallel per point



project/reduce over time



compose multiple time steps

Example Comparison Operations

- Post-classification (color then compare)
  - Alpha Composition (evolution)
  - Color Addition (spatial overlap)
- Pre-classification (compare then color)
  - Numerical operations (integrated data analysis)
    - Min, max, mean, median, etc.
    - Sum, product, difference, etc.
    - Inner product, outer product, etc.
  - Set operations using composition notation (in, out, xor, atop, over, pass, clear) (spatial overlap)

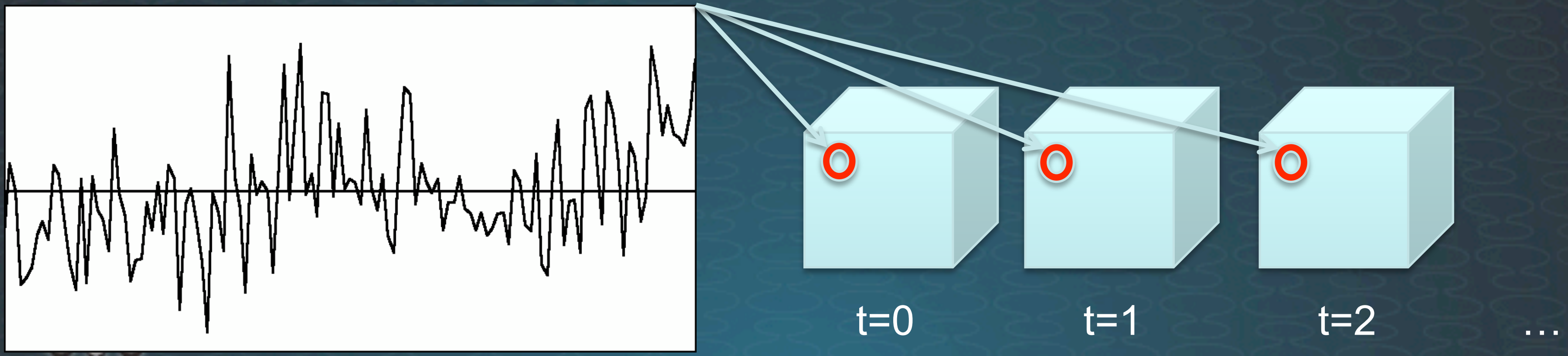
Implementation

- Chronovolumes:
  - for p in all points
    - for t in all time steps in some order
      - output p = reduction operation(output p, p at t)
- Composition:
  - for p in all points
    - output p = apply kernel program at point p (arbitrary selection and composition of time steps)

- Example chronovolume code is provided in chrono.cpp

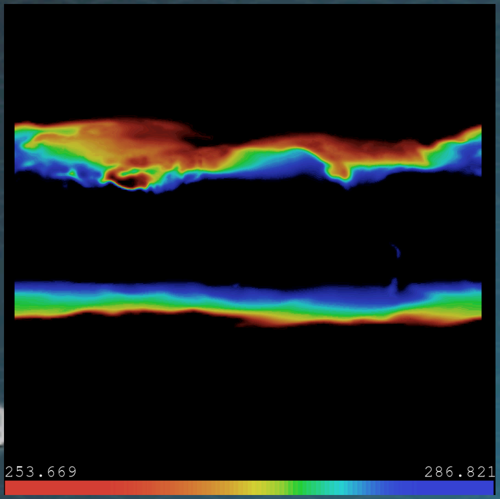
Trend Analysis and Visualization – Time Series/Activity Curve (TAC)

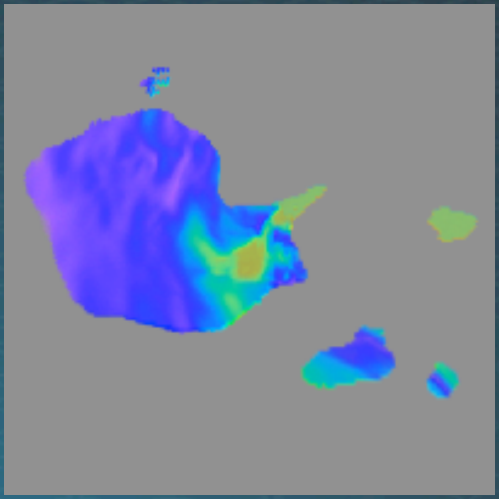
- Quantitative visualization of time-varying data by representing data points as values over time
- Represent data points as time series (activity) curves
  - Coined as TAC vectors by Fang et al.
  - A TAC vector is a data point (point in space) representing data values over time at that point



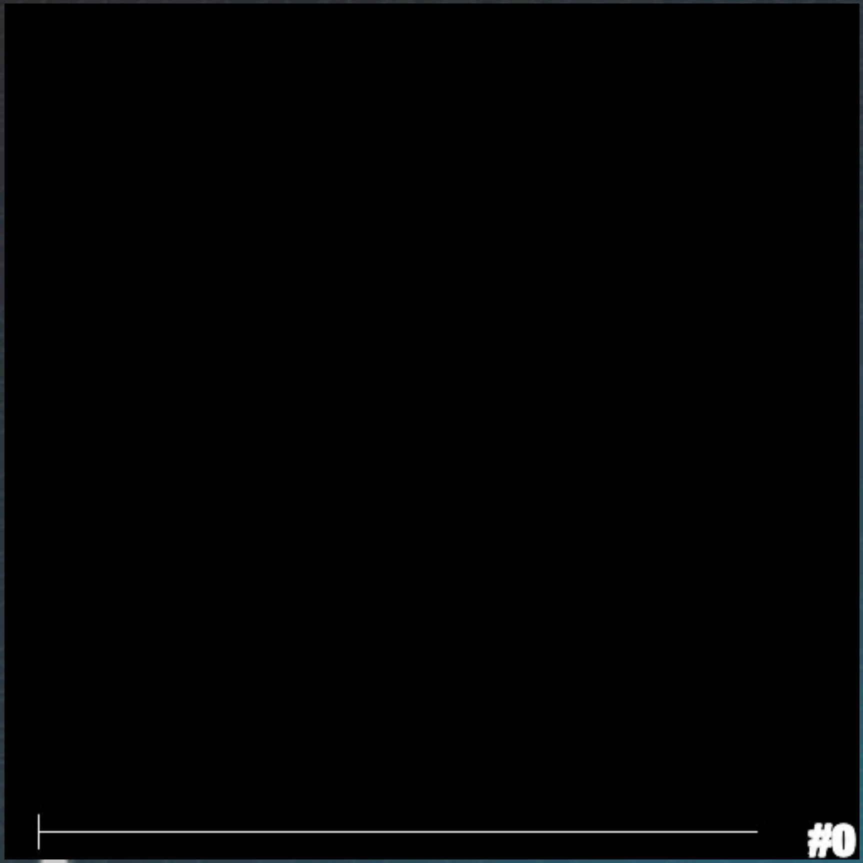
Classifying Features by Trend

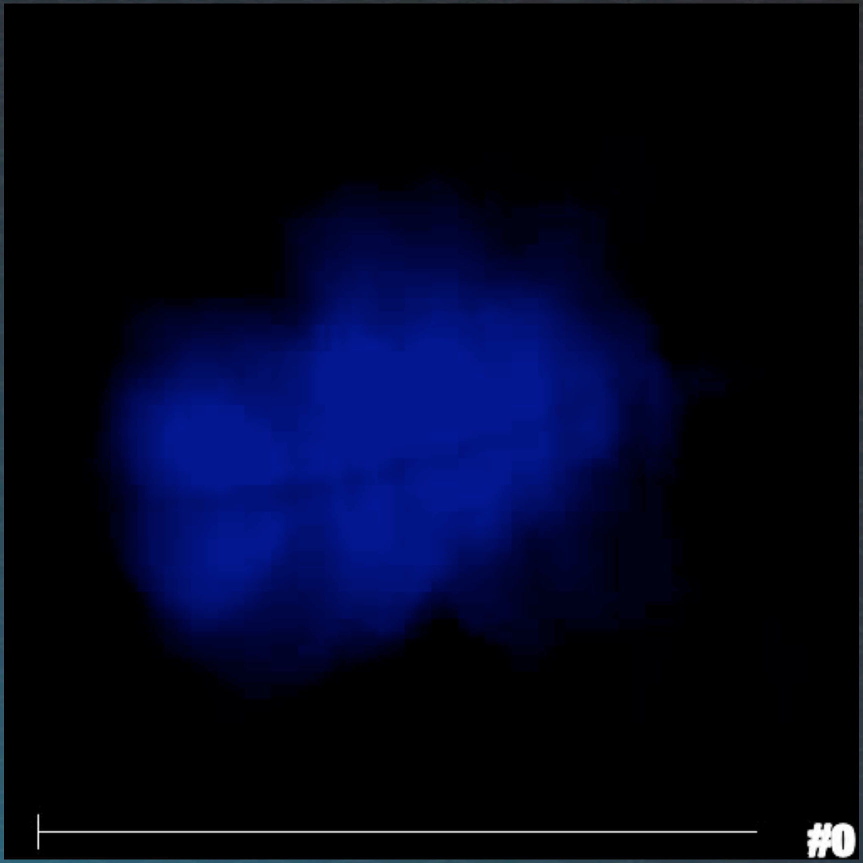
- Our assumption is that features are defined by data points that have similar temporal behavior, (the value change over time is similar) meaning they have a similar TAC
- This can be found computationally through vector clustering of TACs





Animation Rescaling by TAC

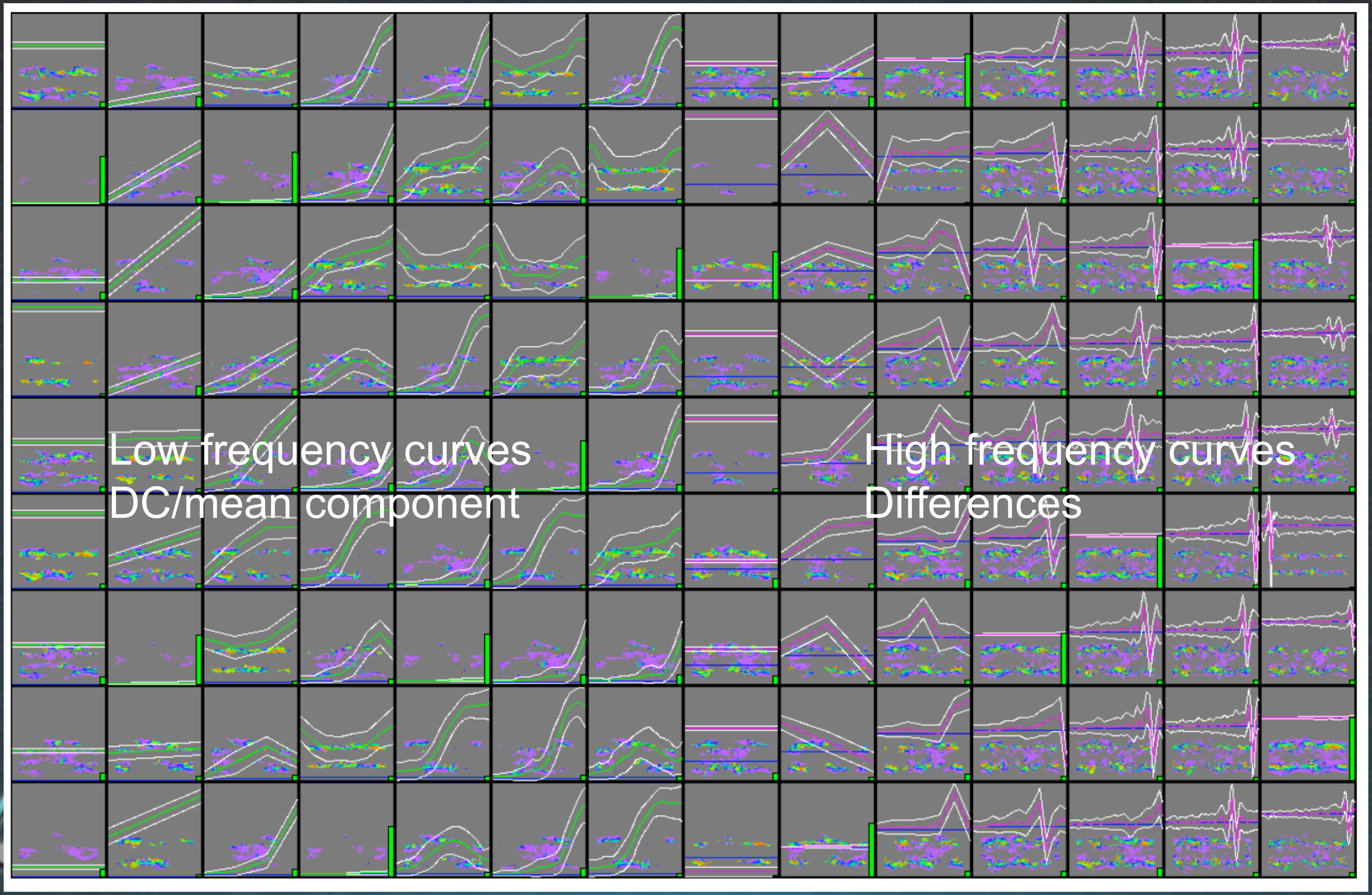




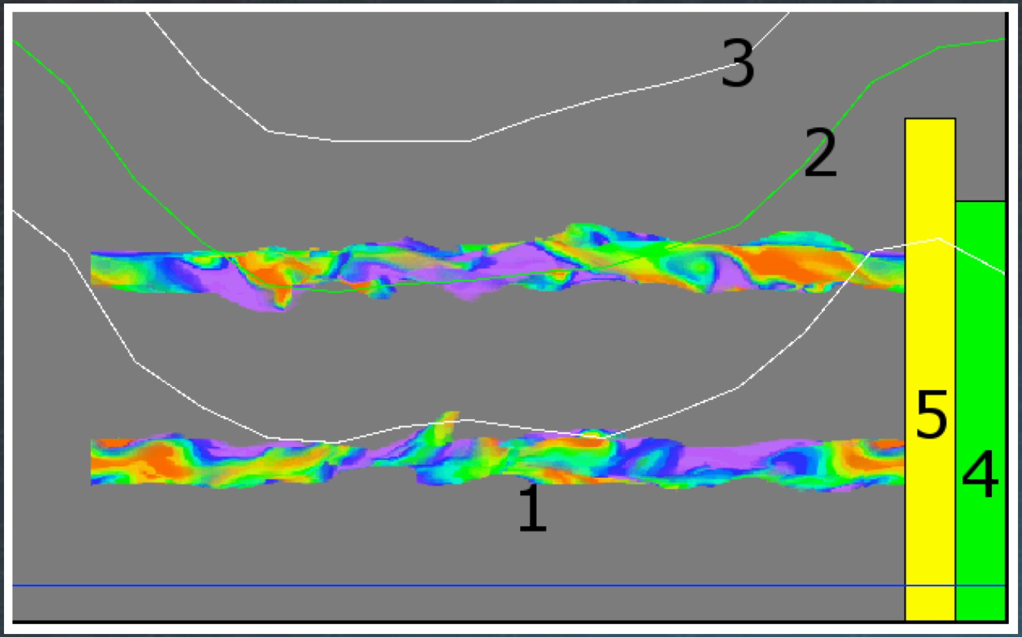
Classes across Time Scales

- Temporal activity can happen at different time scales
  - Short term scale: daily or monthly weather
  - Long term scale: yearly or decadal weather
- Activity classes are clustered by time scale
  - Use filter banks to pass-band filter the TACs into different time scales and then cluster by scale
  - Data points are separately classified in each time scale, thus different trends are identified

Trend Visualization – Visualization Spreadsheet



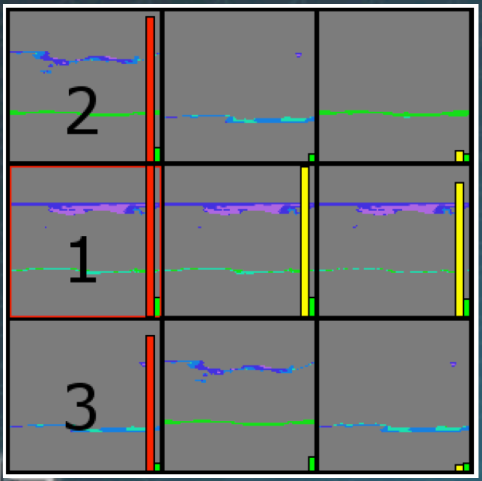
A Cell in the Spreadsheet



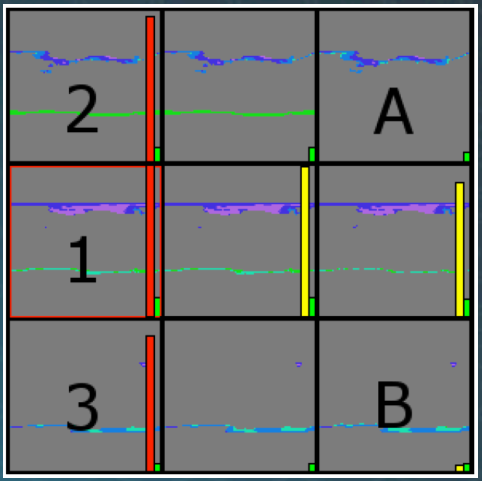
- 1 – thumbnail
- 2 – centroid TAC
- 3 – one stddev value variance from the centroid
- 4 – cluster size (number of points)
- 5 – similarity to a selected cell

Selection and Resorting the Spreadsheet by Relevance

- When a cell is selected, rows and columns are resorted to show relevance to a picked cell (trend cluster)
  - Similar trend clusters are moved closer to the row of a picked cell (doesn't move out of column) based on various distance metrics (centroid, spatial overlap, etc.)

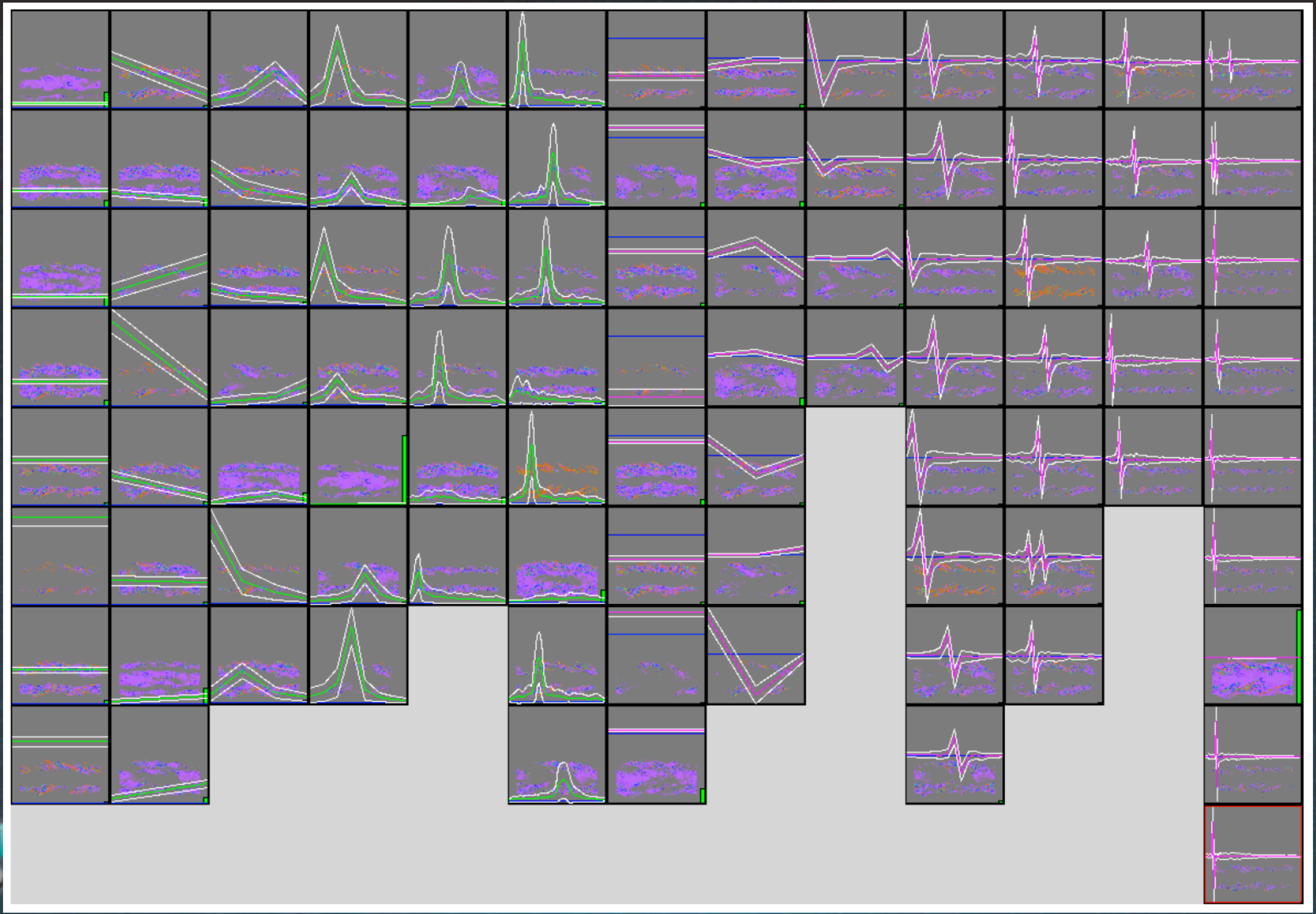


Before selecting cell 1



After selecting cell 1

Spreadsheet Cell Merging and Culling



Implementation

- Represent data as a vector field: rather than scalars of (x, y, z, t) = v, as TAC vectors of (x, y, z) = $<v_0, v_1, v_2, \ldots>$
- Pass-band filter the vectors into different time scales (the paper implementation used wavelets)
- Cluster the vectors using vector clustering by frequency band to separate time scales (k-means, hierarchical, etc.)
- Visualize the clusters in a spreadsheet

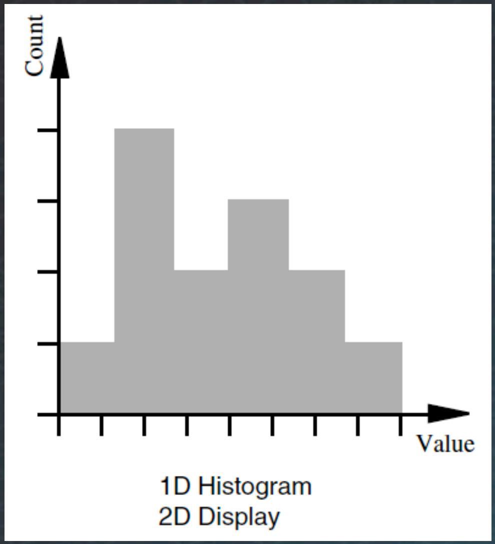
- There isn't an available reference implementation of the spreadsheet and pass-band filtering, but the clustering code can be found in tstf-1.0.tar.gz

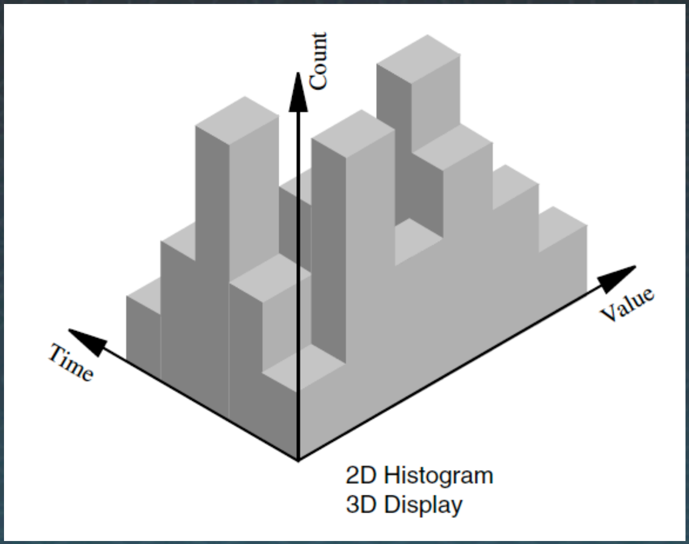
Time Histograms

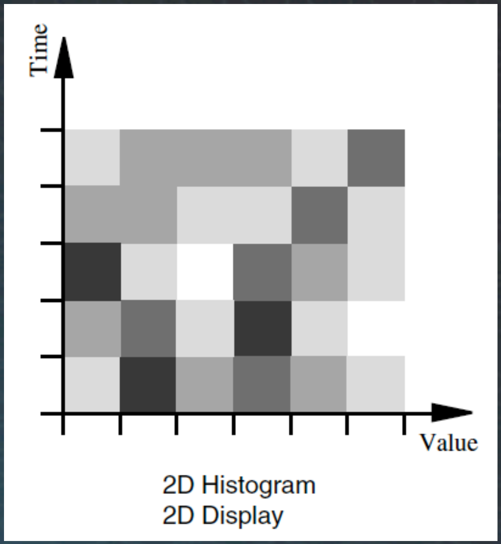
- Trends can also be shown via time histograms
  - They have a better representation of the value distribution over time of a cluster – centroid TAC is just the average trend

- Kosara et al., “TimeHistograms for Large, Time-Dependent Data”
- Akiba et al., “Simultaneous Classification of Time-Varying Volume Data Based on the Time Histogram”

Time Histograms







Typical histogram

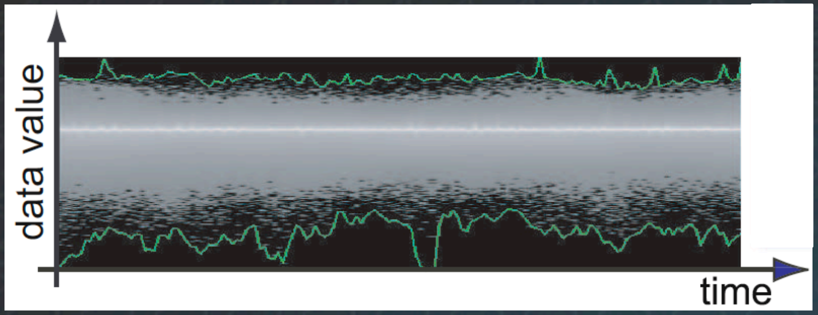
Adding time as a dimension

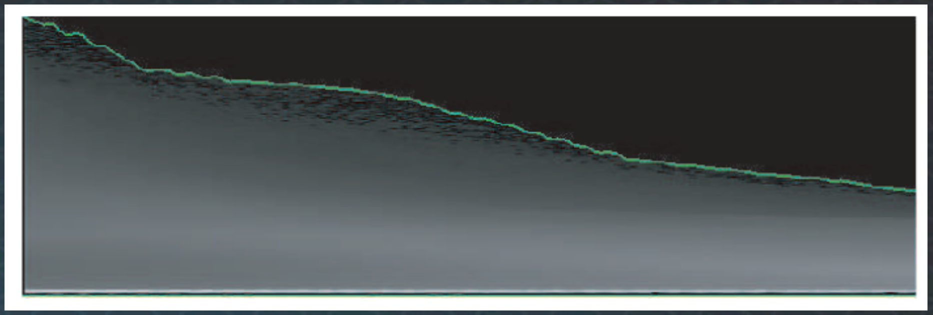
Flattening it
Height = Intensity

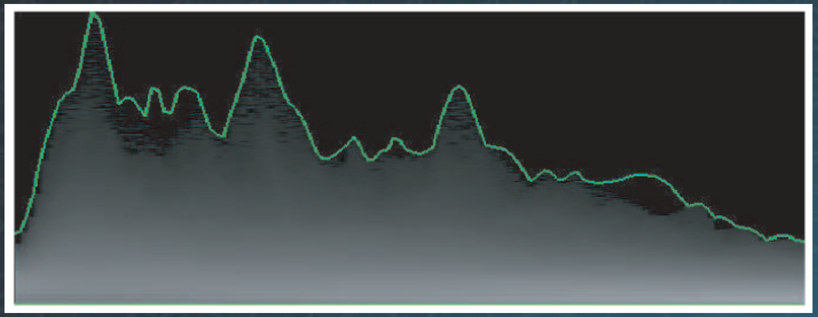
VIS09

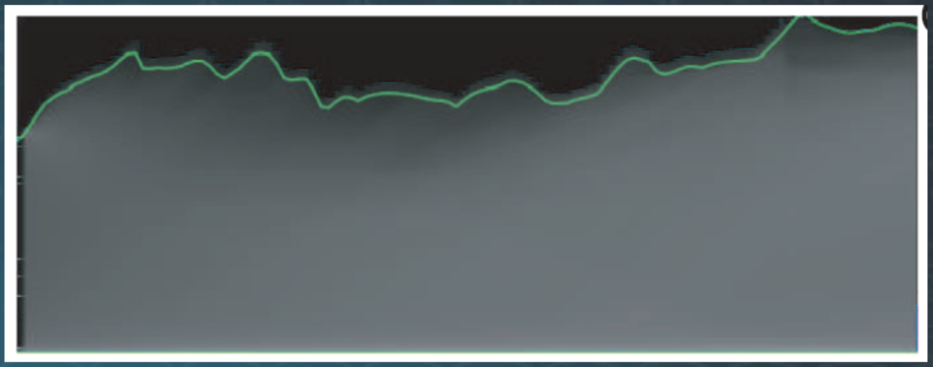
Images from Kosara et al.

Time Histogram Examples









Images from Akiba et al.

Using Time Histograms for Transfer Functions/Classification

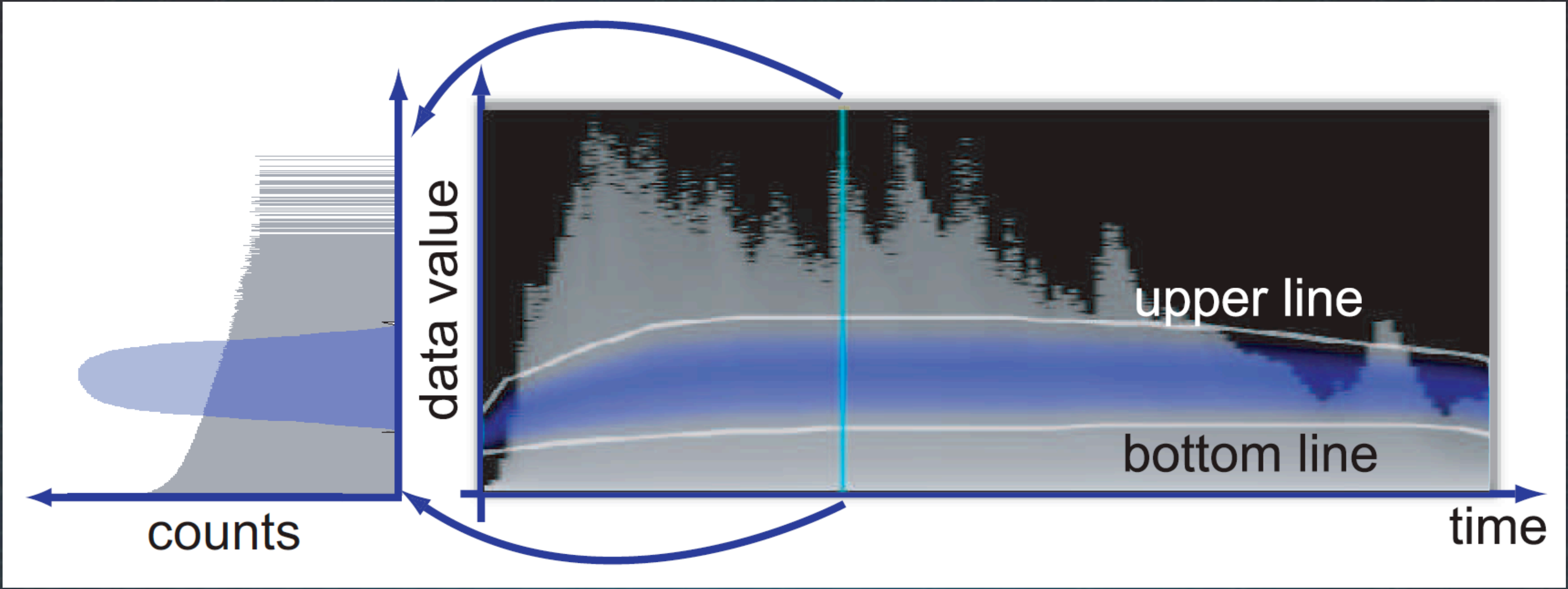


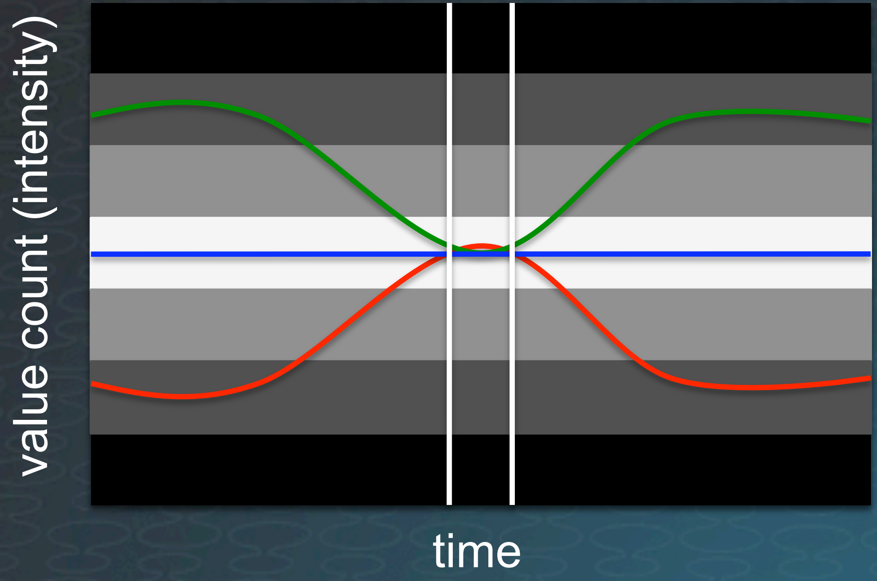
Image from Akiba et al.

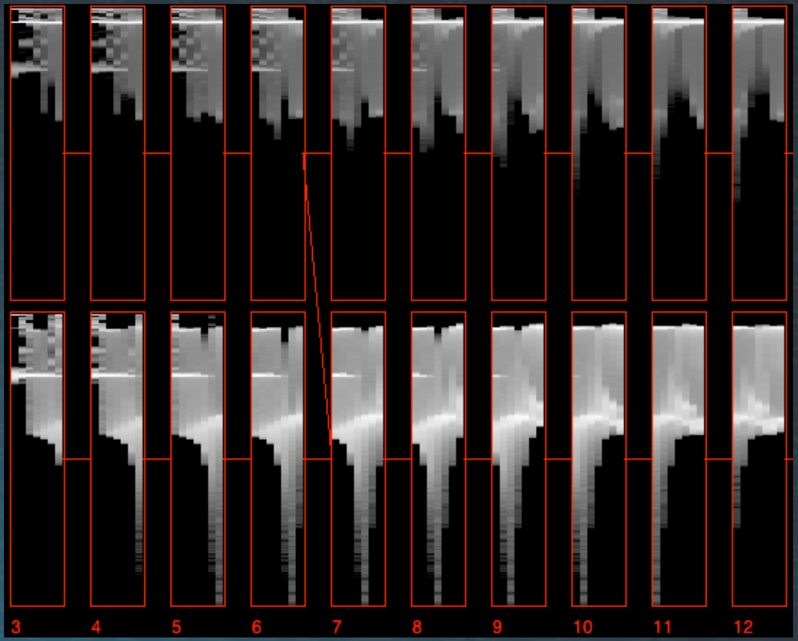
Implementation

- for t in all time steps
  - create a histogram for time step t
- for x in time steps
  - for y in number of histogram bins
    - Render pixel (x, y) as the intensity of bin y in histogram x

- The code in tstf-1.0.tar.gz has examples of creating time histogram images

TAC Clustering Classification compared to Time Histograms

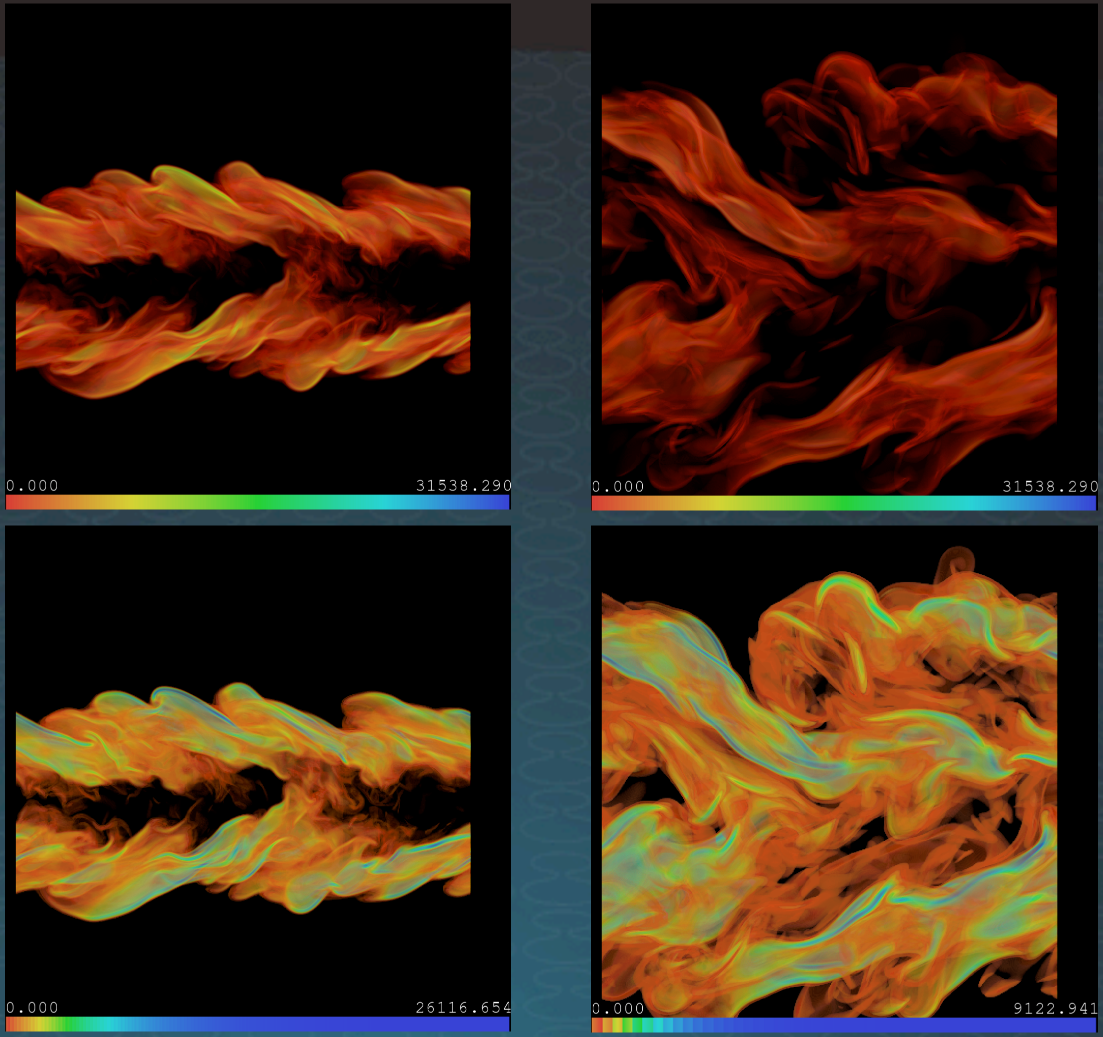




Trend Clustering for Semi-Automatic Time-Series Transfer Functions

Static transfer function

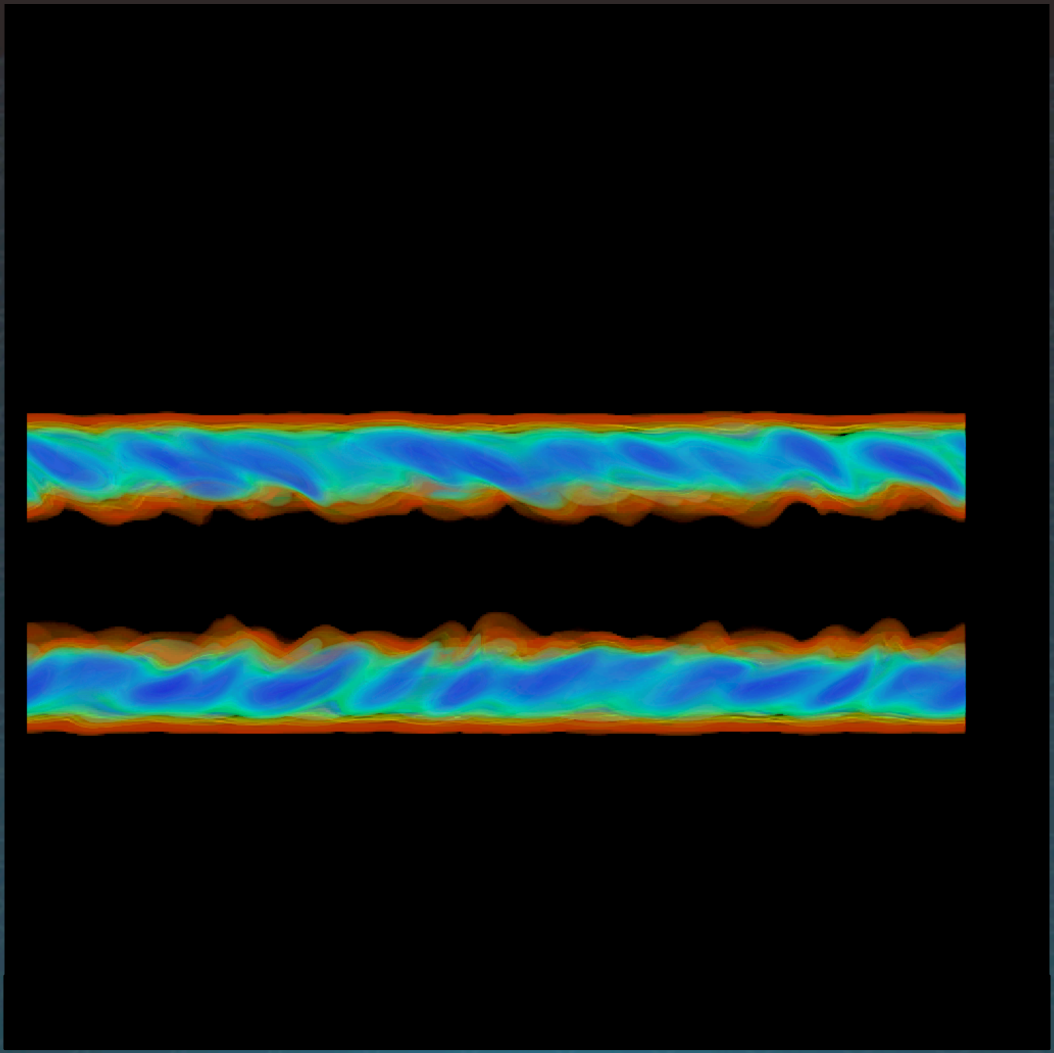
Automatic method



Early time step

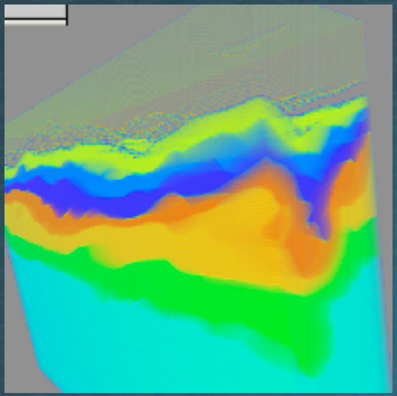
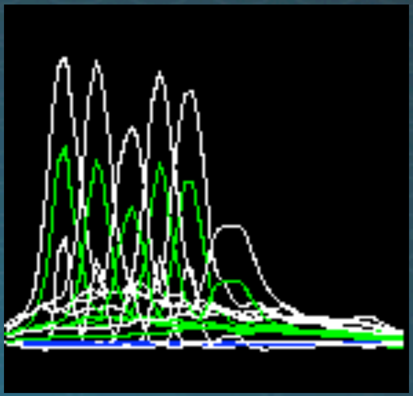
Late time step

Semi-Automatic Time Series Transfer Functions in Animation



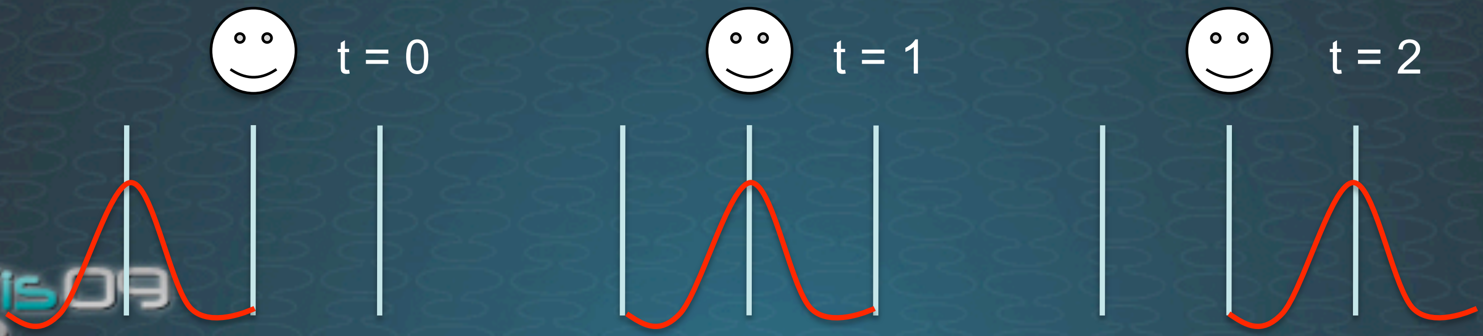
Trend Classification for Moving Features

- Whole time series TAC clustering works for well classifying stationary features that have temporal behavior (climate regions, an earthquake basin)
- For a moving feature/wave, using “normal” TAC clustering, where the clustered time series curves have lengths of the whole time series, it captures the space that a moving wave passes through over time

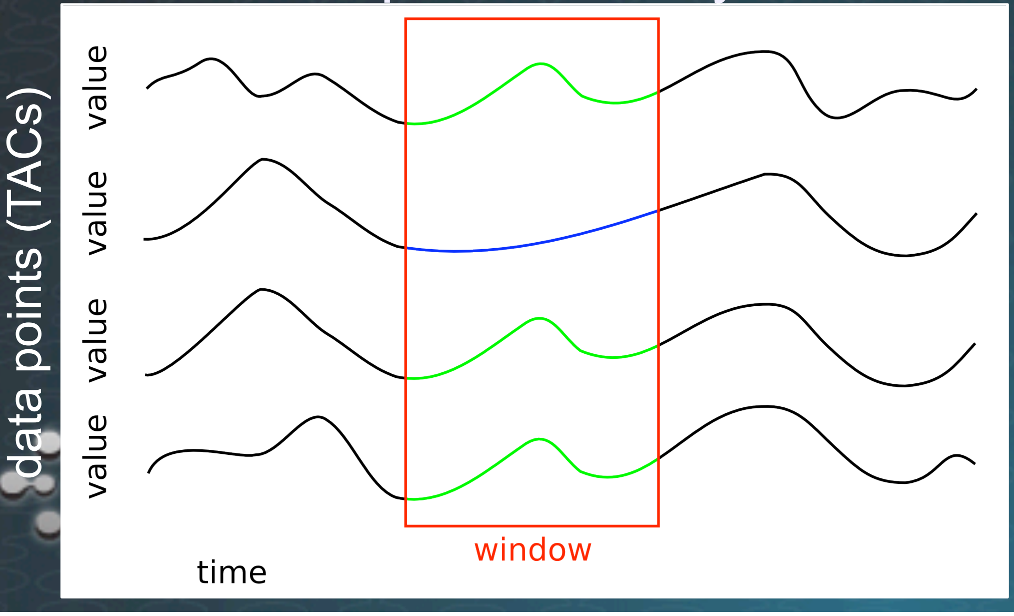
Using this Knowledge to Classify a Moving Wave/Feature

- The data points (spatial area) that comprise a “wave front” have similar behavior for a short period as the wave moves through a region of space
- As a wave moves in space, data points that contain the wave at a point in time, will be similar to a set of data points with the same trend, in near future and near past



Windowed TAC Clustering – Finding Features at a Time Step

- Find similar behavior in the short term: ignore far future and far past similarity
  - Window the TACs (box filter, Gaussian filter, etc.)
  - Similarity is found per time step, by clustering each time step individually

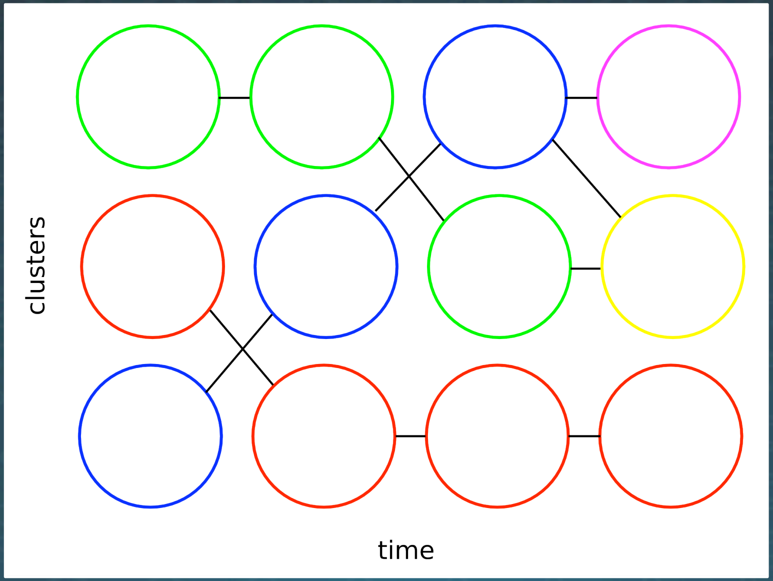


Each time step has a set of clusters which represents similar local activity at that time step – individually, clusters do not represent activity over the entire time range, but just the features/ activities that are occurring in a short time period

Cluster Sequencing –
Connecting the Features over Time

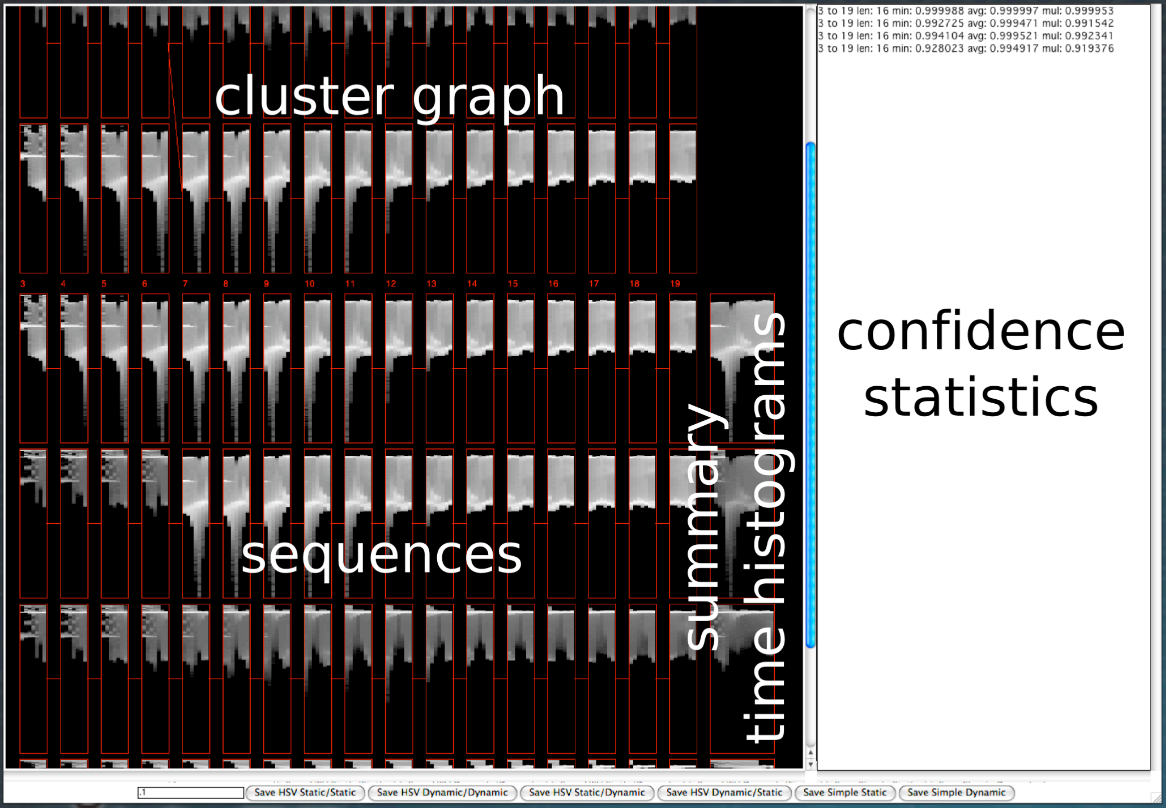
- To connect TAC clusters into an evolving or moving feature, link the clusters over time that are similar/same
  - Measure the similarity between clusters and link them into a sequence by probable evolution

Clusters of data points (the circles) are put into a graph – graph edges are a probability estimate of similarity between clusters over time: low probability edges are culled

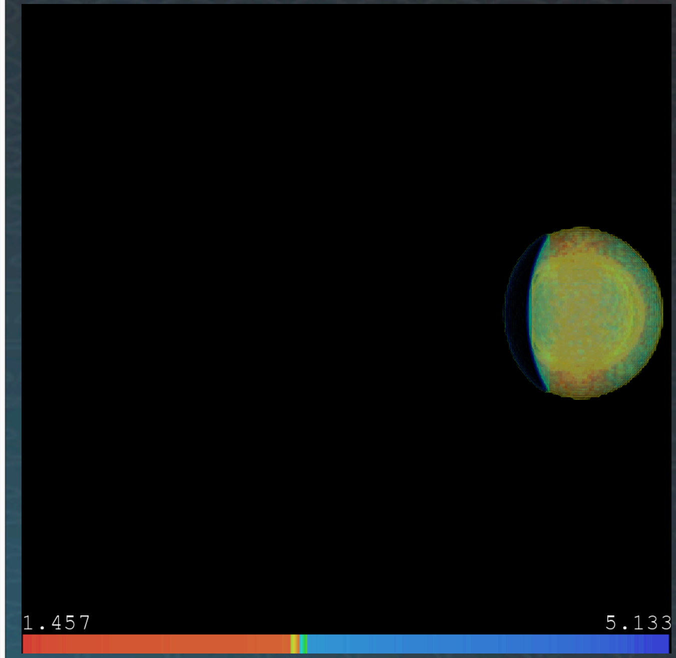


Paths in the graph are classes, which represents a sequence of clusters or the evolution of a value population over time – following local trends over the entire time series

Visualizing the Clusters and Sequences



```
3 to 19 len: 16 min: 0.999988 avg: 0.999997 mul: 0.999953
3 to 19 len: 16 min: 0.992725 avg: 0.999471 mul: 0.991542
3 to 19 len: 16 min: 0.994104 avg: 0.999521 mul: 0.992341
3 to 19 len: 16 min: 0.928023 avg: 0.994917 mul: 0.919376
```



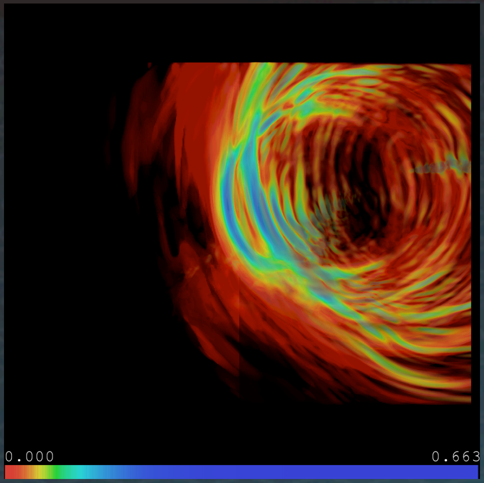
An animation using a selected sequence turned into a transfer function

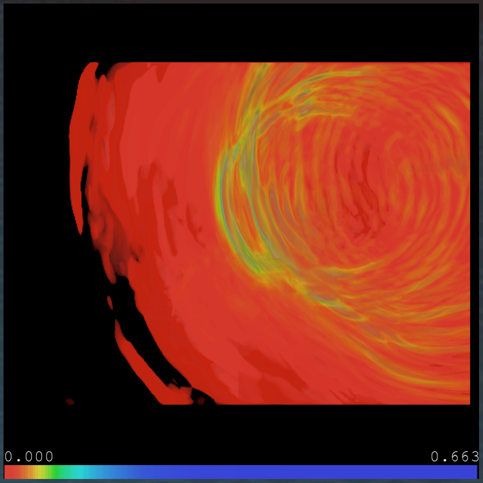
Using the Cluster Sequences for Transfer Functions and Classification

- Sequences are classes of evolving value space features
  - Each cluster in a sequence records value distribution (histogram) of the data points
  - Map the histogram of a time step to a color/alpha map
  - For a dynamic transfer function: use histogram equalization to update the color/alpha map as the value distribution changes over time – i.e., remap the color distribution based on the value distribution shift to maintain visual coherence
- The sequence classification can also be used for isosurfacing, spatial boundaries, etc.

Different Types of Possible Transfer Functions

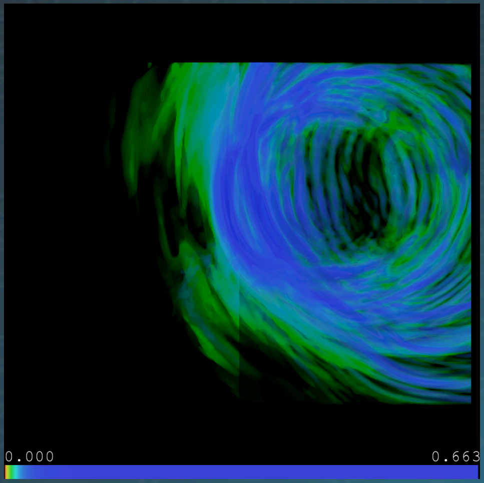
Dynamic Color

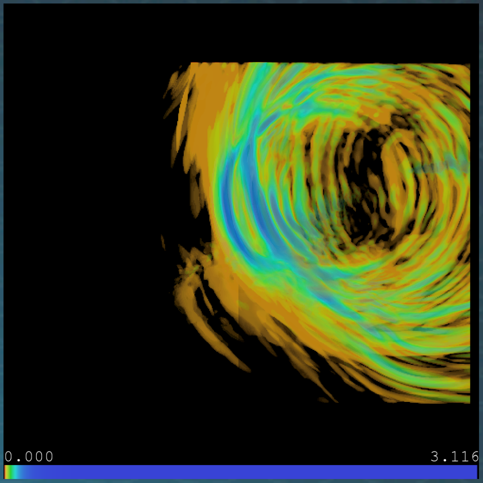




Static = Summed cluster histograms over time = Single map

Static color





Dynamic = Cluster Histogram per time step = Changing Color/alpha map

Dynamic Opacity

Static Opacity

Implementation

- Represent data as a vector field: rather than scalars of (x, y, z, t) = v, as TAC vectors of (x, y, z) = <v_0, v_1, v_2, …>
- for t in all time steps
  - cluster TACs in a small time window around t
- Connect the clusters into a graph, with edges connecting a cluster to all other clusters one time step in the future and past
- A graph edge represents the probability that a cluster is the same value space temporal feature in an adjacent time step – calculate the similarity of adjacent clusters, and remove edges that have low probability (similarity)

Implementation

- Find all paths in the culled graph – paths are sequences or an evolving value space feature
- Visualize the sequences
- Use the sequences in other visualizations
  - To generate a dynamic time-varying transfer function
    - Apply a histogram to color/alpha mapping for one time step
    - Use histogram equalization of the value distribution to update the color/alpha map
- A reference implementation of this work is found in tstf-1.0.tar.gz

For Further Information

- woodring@lanl.gov Jon Woodring
- http://www.cs.utk.edu/~huangj/vis09
- Included are these slides, chrono.cpp, and tstf-1.0.tar.gz
- Woodring and Shen, “Semi-Automatic Time-Series Transfer Functions via Temporal Clustering and Sequencing”
- Woodring and Shen, “Multiscale Time Activity Data Exploration via Temporal Clustering Visualization Spreadsheet”
- Woodring and Shen, “Multi-variate, Time Varying, and Comparative Visualization with Contextual Cues”
- Woodring, Wang, and Shen, “High Dimensional Direct Rendering of Time-Varying Volumetric Data”
- Kosara et al., “TimeHistograms for Large, Time-Dependent Data”
- Akiba et al., “Simultaneous Classification of Time-Varying Volume Data Based on the Time Histogram”

Final Slide

- Jian Huang - huangj@cs.utk.edu
- Wes Kendall - wesleykendall@gmail.com
- Heike Janicke - H.Jaenicke@swansea.ac.uk
- Chaoli Wang - chaoliw@mtu.edu
- Jon Woodring - woodring@lanl.gov

- www.cs.utk.edu/~huangj/vis09
  - Slides will be uploaded soon